Quantum Neural Nets

Michail Zak and Colin P. Williams

Center for Space Microelectronics Technology
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

Abstract

The capacity of classical neurocomputers is limited by the number of classical degrees of freedom which is roughly proportional to the size of the computer. By contrast, a hypothetical quantum neurocomputer can implement an exponentially larger number of the degrees of freedom within the same size. In this paper an attempt is made to reconcile linear reversible structure of quantum evolution with nonlinear irreversible dynamics for neural nets.

1. Introduction

The competition between digital and analog computers, i.e., between computations and simulations, has a long history. During the last fifty years, the theory of computation has been based, implicitly, upon classical physics as idealized in the deterministic Turing machine model. However, despite the many successes of digital computers, the existence of so called hard problems has revealed limitations on their capabilities, since the computational time for solving such problems grows exponentially with the size of the problem.

" It was well understood that one possible way to fight the "curse" of the combinatorial explosion is to enrich digital computers with analog devices. In contradistinction to a digital computer, which performs operations on numbers symbolizing an underlying physical process, an analog computer processes information by exploiting physical phenomena directly. It is this problem solving via direct simulation that allows an analog approach to reduce the complexity of the computations significantly. This idea was stressed by Feynman [1] who demonstrated that the problem of exponential complexity in terms of calculated probabilities can be reduced to a problem of polynomial complexity in terms of simulated probabilities. Conceptually, a similar approach can be applied to the whole class of NP-complete problems. Indeed, the theory of computational complexity is an attribute of digital approach to computations. At the same time, in principle, one can find such a physical phenomenon whose mathematical description is equivalent to those of a particular NP-complete problem. Them, incorporating this phenomenon into an appropriate analog device, one can simulate the corresponding NP-complete problem. But is it possible, in general, to find a new mathematical formulation for any intractable problem in such a way that it becomes tractable? Some experts in computational complexity believe that, in the spirit of the Gödel theorem, there always exists computational problems such that every mathematical formulation that captures the essence of the problem is intractable [2]. At this step, we cannot prove or disprove this statement.

There is another class of problems for which simulations are superior over computations. In contradistinction to NP-complete problems whose complexity is in an exponentially large number of simple computations, these problems have complex and sometimes, partially unknown analytical structure. Simulations of solutions to such problems are based upon a black-box approach when unknown components of the model are found in the course of a trial-and-error learning process. A typical representative of a corresponding analog device implementing black-box-based simulations is a neurocomputer where unknown (learnable) parameters are incorporated in the form of synaptic interconnections between dynamical units called “neurons”. However, usually analog computers are associated with certain limitations such as the lack of universality, slow performance, and low accuracy, and this is the price to be paid for certain advantages of simulations. A partial success in development of a universal analog device is associated with neurocomputers which are based upon massively parallel adaptive dynamical systems modeled on the general features of biological neural networks that are intended to interact with the object of the real world in the same way the biological systems do. However, the capacity of the neurocomputers is roughly proportional to the size of the apparatus, and that limits actual power significantly.

A second way to fight a curse of dimension is to utilize a non deterministic approach to computations. This approach is associated with the Monte Carlo method introduced by N. C. Metropolis and S. M. Ulam in 1940. The idea of this method is based upon the relationships between the probabilistic characteristics of certain stochastic processes and solutions to some deterministic problems such as values of integrals, solutions to differential equations, etc. The strength of the method is that its error does not depend on the number of variables in the problem, and therefore, if applicable, it breaks the curse of dimension. The effectiveness of the Monte-Carlo approach is inversely proportional to the smoothness parameter that characterizes the degree of correlation within the input data. However, the Monte-Carlo method is not the only way to apply nondeterminisism for computations. There is a class of so-called randomized algorithms that are effective for combinatorial problems. In general, a randomized strategy for this kind of problem is useful when there are many ways in which an algorithm can proceed, but it is difficult to determine a way that is guaranteed to be good. In particular, if the benefits of good choices outweigh the costs of bad choices, a random selection of good and bad choices can yield a good algorithm.

In general, the theory of computational complexity proves that polynomial time nondeterministic algorithms are more powerful than polynomial time deterministic ones. For example, certain nondeterministic algorithms are known that can probably find a solution within guaranteed polynomial time. While other nondeterministic algorithms are known that can certainly find a solution within probably polynomial time. Thus randomized algorithms defeat the curse of dimension by trading complexity for correctness or completeness. However, the main limitation of the whole non deterministic approach is in the generation of random numbers: the generators are slow and not always reliable (i.e. the sequence of numbers that they produce may harbor hidden correlations that no truly random sequence would possess) . That is why the concept of a quantum computer became so attractive: its analog nature is based upon physical simulations of quantum probabilities, and at the same time, it is universal (at least for modeling physical world).

Although the development of the quantum-mechanical device is still in progress, a new quantum theory of computations has been founded [3], [4],[8]. This theory suggests that there is a second fundamental advantage of the hypothetical quantum computer which is based upon the wave properties of quantum probabilities: a single quantum computer can follow many distinct computational paths all at the same time and produce a final output depending on the interference of all of them. This particular property opened up a new chain of algorithms that solve in polynomial time such hard problems as factorization and discrete log, i.e., the problems that are believed to be intractable on any classical computer.

In order to clarify the connection between quantum algorithms and combinatorial optimization, consider n binary variables x = xl, $x_2, \dots x_n ; x_i \dots \in \{0,1\}$, and combine them into larger number of variables as all possible products of n old variables:

$$y = x_1 \otimes x_2 \otimes \dots \otimes x_n = x_1 x_2 \dots x_m, x_2 x_1 \dots x_{m.} \qquad etc.$$

The number of these new variables is N = 2“ . In many practical applications, a function to be optimized is defined at a set of the new variables. For instance, in the course of a spacecraft design, the optimal placement of sensors requires to minimize the cost function which depends upon 2“ values of y, since each of n assigned places can be equipped or not equipped by a sensor. Since the number of possible assignments grows exponentially with the number of placements, it appears that the time required to solve this problem must also grow exponentially (in the worst case) even if a single computation of each of 2“ values of the cost function is trivial. Actually it is this property which makes the problems of combinatorial optimization intractable by classical computing.

There is a striking similarity between the structure of combinatorial problems and some special properties of quantum evolution, namely, the property of quantum entanglement. This property follows from the fact that if two unitary matrices $U,$ and U_2 are solutions to the Schrodinger equation, their tensor product $U, \otimes U_2$ *will* be also the solution to it. Therefore, with an input of n binary variables of the type x, one can obtain 2“ variables y as an output in one computational run. In other words, the transition from n basic variables x to 2^n combinatorial variables y are carried out by the laws of Nature, and that is the analog foundation of quantum computing. Unfortunately, Nature also imposes severe restrictions on the amount of information that can be extracted from the superposition of y answers. In particular, a direct measurement will yield only one answer, although more clever measurement schemes can reveal certain joint properties of all the y answers. The technique of quantum parallelism relies upon the use of the latter type of measurements.

Actually the transition from x to y is carried out by n of 2 x 2 identity matrices $I^{(i)}$ as following:

$$y_{j_1 \dots j_n} = \sum I^{(1)}_{j_1 i_1} \dots I^{(n)}_{j_n i_n} x_{i_1 \dots i_n}$$

Replacing identity matrices by non-identical unitary matrices $S^{(,)}$ one finds a new variable $z_{j_1 \dots j_n}$ which is combined of weighted sums of all the components of the variable y, and that is due to another fundamental property of quantum mechanics: the interference of probabilities (which is postulated).

If the matrices $S^{(,)}$ are chosen such that the variable z is equal to the cost function, then the computation is accomplished: the output contains all the 2“ values of the cost function. However, in order to find the optimal value of the combinatorial variable $Y = y_o$, one has to impose an additional constraint UPOn the matrices $S^{(,)}$> namely: the weight coefficient of y_o must dominate over other weight coefficients in order to detect this optimal value in a few number of measurements, and this constraint is, probably, the toughest. Hence quantum computing does not allow iterations, feedbacks, or any other types of control over the computational process: one must get the solution at once, or he does not get it at all.

Thus, there are at least two areas where the quantum computer is expected to be superior over the classical one: quantum mechanics (due to simulation of quantum probabilities), and some specific combinatorial problems linked to operation research (due to interference entanglement and of quantum probabilities).

In this paper an attempt is made to combine the power of quantum computing and the dynamical complexity of neural nets. There are at least three reasons for such combinations. Firstly, it will represent a universal analog device with a built-in random number generator. Secondly, its capacity will be exponentially larger than those of a classical neurocomputer due to the superposition and entanglement effects. Thirdly, it will introduce iterations in quantum computing.

The main challenge of the approach is in reconciliation of linear reversible quantum evolution and nonlinear irreversible dynamics of neural nets.

2. Neural Net as Dynamical System

A neural net as a nonlinear dissipative dynamical system can be represented by the following set of ordinary differential equations:

$$\tau_i \dot{x}_i = -x_i + \sigma\left(\sum_j T_{ij} x_j\right), \qquad \tau_i > 0 \qquad (1)$$

where x, are state variables, or mean soma potentials, characterizing the neuron activities, T_{ij} , are constant control parameters representing the weights of synaptic interconnections, τ_i are suitable time constants, and $\sigma(\cdot)$ is a sigmoid function having a saturated non linearity [usually o(x) = tanh βx, where $\beta = const > 0$ is an additional control parameter].

An invariant characterizing the local dissipativity of the system (1) is expressed explicitly via its parameters:

$$\operatorname{div} \dot{x} = \sum_i \frac{1}{\tau_i}\left(-1 + \frac{\beta T_{ii}}{\cosh^2 \sum_j T_{ij} x_j}\right) \qquad (2)$$

A necessary (but not sufficient) condition that the system (1) has attractors is that there are some domains in phase space where the invariant (2) is negative.

If the matrix T_{ij} is symmetric

$$T_{ij} = T_{ji} \qquad (3)$$

then equation (1) can be represented in the form of a gradient system, and therefore it can have only static attractors. In the basin of a static attractor, the invariant (2) must be negative.

Since the system (1) is nonlinear, it can have more than one attractor; consequently, in some domains of phase space, the invariant (2) may be positive or zero.

Equation (1) presents the neural net in its "natural" form in the sense that x_i and T_{ij} correspond to physical parameters: neuron potentials and synaptic interconnections, respectively. However, it is important to emphasize that the relationship between the invariant of the "vector" x_i and the "tensor" T_{ij} are not preserved by the coordinate transformation, i.e., equation (1) does not possess an invariant tensor structure. Consequently, the column xi and the matrix T_{ij} cannot be treated as a vector and tensor, respectively.

In most applications, e.g. pattern recognition, optimization, decision-making, control, associative memory, generalization, the neural nets performance is associated with convergence to attractors The locations of attractors and their basins in phase space can be prescribed by an appropriate choice of the synaptic weights T_{ij}, i.e., by solving inverse dynamical problems. However, since dimensionality of neural nets is usually very high (in biological systems it is of order of 10^{11} with the number of synaptic interconnections of the order of 10^{15}), the straightforward analytical approach can be very expensive and time consuming. An alternative way to select synaptic weights in order to do specific tasks was borrowed from biological systems. It is based upon iterative adjustments of T_{ij} as a result of comparison of the net output with known correct answers (supervised learning) or as a result of creating of new categories from the correlations of the input data when correct answers are not known (unsupervised learning). Actually the procedure of learning is implemented by another dynamical system with the state variables T_{ij} which converges to certain attractors representing the desired synaptic weights T^{Y}_{ij}.

Equation (1) represents a so-called continuously updated neural net. Its discrete version is modeled by a corresponding contracting nonlinear map whose dynamical behavior, in principle, is similar to those of Eq. (1). In the simplest form such a map can be written in a McCulloch-Pitts form [5]:

$$x_i(t+1) = sgn \sum T_{ij} x_j(t) \tag{4}$$

where the sign function (sgn) plays the role of the sigmoid σ.

By replacing sgn in (4) with a stochastic rule:

$$x_i(t+1) = S \sum T_{ij} x_j(t) \tag{5}$$

$$S = \begin{cases} +\ 1 \text{ with probability } f\left(\sum T_{ij}x_j\right) \\ -1 \text{ with probability } 1- f\left(\sum T_{ij}x_j\right) \end{cases} \qquad (6)$$

one arrives at a stochastic version of neural nets, while the actual implementation of the stochastic rule (6) is still to be based upon a random number generator.

The basic limitation of deterministic or stochastic classical neurocomputers is in their restricted capacity which is proportional to the size of the computer. This limitation becomes obvious when neurocomputer is compared with a human brain: there are 10” of parallel units in a human brain while neural chips made so far contain of the order of 10’ units, which is too few for most practical applications, [5].

3. Quantum Model of Evolution

A state of a quantum system is described by a special kind of time dependent vector $|\psi>$ with complex components called amplitudes. It will help to make the correspondence with Markov chains clearer if we define this vector in bra form:

$$\{a_0 a_1 \ldots a_n\} = \langle\psi \qquad (7)$$

If unobserved, the amplitudes evolve in accordance with Schrödinger’s equation:

$$i\hbar\frac{da_k}{dt} = \sum_{\ell} H_{k\ell}a_{\ell} \qquad (8)$$

which is linear and reversible.

Here $H_{k\ell}$ is the Hamiltonian of the system, $i = \sqrt{-1}$ and $\hbar = 1.0545 \text{ x } 10^{-34} JS$.

The solution to Eq. (8) can be written in the following form:

$$\{\%(0..., \ \% \ (t)\} = \{a_0(0), \ldots, a_n(0)\}U^* \qquad (9)$$

where U is a unitary matrix uniquely defined by the Hamiltonian:

$$U = e^{-iHt/\hbar} \quad , \; UU^* = I \qquad \text{(lo)}$$

After m equal time steps At

$$\{a_0(m\Delta t), \dots a_n(m\Delta t) = \{a_o(0) \dots a_n(0)\} U^{*,m} \qquad (11)$$

the transformation of the amplitudes formally looks like those of the transition probabilities in Markov chains. However, there is a fundamental difference between these two processes: in Eq. (11) the probabilities are represented not by the amplitudes, but by squares of their modules:

$$P = \left\{ |a_o|^2, \dots |a_n|^2 \right\} \qquad (12)$$

and therefore, the unitary matrix U is not a transition probability matrix.

It turns out that this difference is the source of so called quantum interference which makes quantum computing so attractive. Indeed, due to interference of quantum probabilities:

$$P = |a_1 + a_2|^2 \neq P_1 + P_2 \qquad (13)$$

each element of a new vector $a_i(m\Delta t)$ in Eq. (11) will appear with the probability $a_i^{\,2}$ which includes all the combinations of the amplitudes of the previous vector.

4. Quantum Collapse and Sigmoid Function

As mentioned above, neural nets have two universal features: dissipativity and nonlinearity. Due to dissipativity, a neural net can converge to an attractor and this convergence is accompanied by a loss of information. But such a loss is healthy: because of it, a neural net filters out insignificant features of a pattern vector while preserving only the invariants which characterizes its belonging to a certain class of patterns. These invariants are stored in the attractor, and· Hereford, the process of convergence performs generalization: two different patterns which have the same invariants will converge to the same attractor. Obviously, this convergence is irreversible.

The nonlinearity increases the neural net capacity: it provides many different attractors including static, periodic, chaotic and ergodic, and that allows one to store simultaneously many different patterns.

Both dissipativity and nonlinearity are implemented in neural nets by the sigmoid (or squashing) function discussed in Section 2. It is important to emphasize that the only qualitative properties of the sigmoid function are those which are important for the neural net performance, but not any specific forms of this function. Can we find a qualitative analog of a sigmoid function in quantum mechanics? Fortunately, yes: it is so called quantum collapse which occurs as a result of quantum measurements. Indeed, the result of any quantum measurement is always one of the eigenvalues of the operator corresponding to the observable being measured. In other words, a measurement maps a state vector of the amplitudes (7) into an eigenstate vector.

$$\{a_o a_{1\ldots} \ldots a_n\} \to \left\{ 0,0\ldots \underset{`r,}{1}\ .\ ..00 \right| \tag{14}$$

While the probability that this will be the i^{th} eigenvector is:

$$p_i = a_i{}^{2} \tag{15}$$

The operation (14) is nonlinear, dissipative, and irreversible, and it can play the role of a natural “quantum” sigmoid function.

5. Quantum Neural Net Architectures

Let us introduce the following sequence of transformations for the state vector (7):

$$|\psi(0)\rangle \rightarrow U|\psi(0)\rangle \rightarrow \sigma_1\{U|\psi(0)\rangle\} = |\psi(t+1)\rangle \qquad (16)$$

which is a formal representation of (14) with σ_1 denoting a “quantum” sigmoid function.

In order to continue this sequence, we have to reset the quantum device considering the resulting eigenstate as a new input. Then one arrives at the following neural net:

$$a_i(t+1) \quad = \quad \sigma_1\{\sum U_{ij} a_j(t)\} \qquad\qquad i = 1,2\ldots n \qquad (17)$$

which has the form similar to (5).

However, there are two significant differences between the quantum (17) and classical (5) neural nets. Firstly, in Eq. (17) the randomness appears in the form of quantum measurements as a result of the probabilistic nature of the quantum mechanics, while in (5) a special device generating random numbers is required. Secondly, if the dimension of the classical matrix T_{ij} is N x N , then within the same space one can arrange the unitary matrix U (or the Hamiltonian H) of dimension 2’” x 2^N exploiting quantum entanglement,

One should notice that each non-diagonal element of the matrix H may consist of two independent components: real and imaginary. The only constraint imposed upon these elements is that H is the Hermitian matrix, i.e.,

$$H_{ij} = \overline{H}_{ji}, \qquad (18)$$

and therefore, the n x n Hermi tian mat rix has n^2 independent components.

So far the architecture of the neural net (17) was based upon one measurement per each run of the quantum device. However, in general, one can repeat each run for ℓ times $(\ell \leq n)$ collecting ℓ independent measurements. Then, instead of the mapping (14), one arrives at the following best estimate of the new state vector:

$$\{a_0, \cdots, a_n\} = \{0, \ldots \underset{\underset{i_1}{\uparrow}}{\frac{1}{\sqrt{\ell}}}, \ldots, \ldots, \underset{\underset{i_\ell}{\uparrow}}{\frac{1}{\sqrt{\ell}}}, \ldots\} \tag{19}$$

while the probability that the new state vector has non-zero i_k^{th} component is

$$p_{ik} = \alpha_{ik}^{2} \tag{20}$$

Denoting the sigmoid function corresponding to the mapping (19) as σ_ℓ, one can rewrite Eq. (17) in the following form:

$$a_i(t+1) = \sigma_\ell\{U_{ij}a_j(t)\} \tag{21}$$

The next step in complexity of the quantum neural net architecture can be obtained if one introduces several quantum devices with synchronized measurements and resets:

$$a_i^{(1)}(t+1) = \sigma_{\ell_1\ell_2}\{U_{ij}^{(1)}a_j^{(1)}(t)\} \quad i = 1,2 \ldots n_1 \tag{22}$$

$$a_i^{(2)}(t+1) = \sigma_{\ell_2\ell_1}\{U_{ij}^{(2)}a_j^{(2)}(t)\} \quad i \neq 1,2 \ldots n_2 \tag{23}$$

Here the sigmoid functions $\sigma_{\ell_1\ell_2}$ and $\sigma_{\ell_2\ell_1}$, map the state vectors into a weighted combinations of the measurements:

$$\{a_1^{(1)} \ldots a_n^{(1)}\} \to \frac{\alpha_{11}a_{\ell_1}^{(1)} + \alpha_{12}a_{\ell_2}^{(2)}}{\left|\alpha_{11}a_{\ell_1}^{(1)} + \alpha_{12}a_{\ell_2}^{(2)}\right|} \tag{24}$$

$$\{a_1^{(2)} \ldots a_n^{(2)}\} = \frac{\alpha_{21}a_{\ell_1}^{(1)} + \alpha_{22}a_{\ell_2}^{(2)}}{\left|\alpha_{21}a_{\ell_1}^{(1)} + \alpha_{22}a_{\ell_2}^{(2)}\right|} \tag{25}$$

where $a_{\ell_1}^{(1)}$ and $a_{\ell_2}^{(2)}$ are the result of measurements presented in the form (19), and $\alpha_{11,}\alpha_{12,}\alpha_{21}$ and $\boldsymbol{a}_{22}$ are constants.

Thus, Eqs. (22) and (23) evolve independently during the quantum regime, i.e., in between two consecutive measurements; however, during the measurements and resets they are coupled via the Eqs. (24) and (25).

It is easy to calculate that the neural nets (17), (21) and (22), (23) operate with patterns whose dimensions are n, $n(n-1)(n-\ell)$, $n_1(n_1-\ell_1)$ and $n_2(n_2-1)\dots(n_2-\ell_2)$, respectively.

In a more general architecture, one can have K-parallel quantum devices U, with ℓ_i consecutive measurements M_i for each of them (i= 1,2...k), see Fig. 1.

6. Maximum Likelihood Dynamics.

Let us turn to the simplest version of a quantum neural net (17). As pointed out above, its performance is non-deterministic in a sense that each independent run of Eqs. (17) may lead to a different trajectory. However, in order to understand better the nonlinear structure of Eq. (17), we will introduce the best estimate, or the maximum likelihood trajectory by replacing the highest probability term in the output state by one. Choosing, for simplicity, a unitary matrix with real components:

$$U=\begin{pmatrix} 0.858726 & 0.387195 & -0.17004 & 0.289405 \\ 0.179855 & -0.801066 & 0.0518717 & 0.568555 \\ -0.362639 & 0.144341 & -0.832118 & 0.394003 \\ -0.314219 & 0.433058 & 0.525334 & 0.661628 \end{pmatrix} \tag{26}$$

one can verify that any initial state which is sufficiently close to the state {0001} will be attracted to it, and therefore, the eigenstate {0001} is a static attractor. In the same way one can find other static attractors, for instance

{1000] {0100} {OO1O}*{I 100} etc (27)

Another unitary matrix

$$U = \begin{bmatrix} -0.377565 & 0.554112 & -0.741892' \\ -0.70484 & 0.347627 & 0.618349 \\ -0.600537 & -0.756383 & -0.259309, \end{bmatrix} \qquad (28)$$

produces periodic attractors:

$$\{100\} \rightarrow \{010\} \rightarrow \{001\} \rightarrow \{100\} \text{ etc.}$$

Thus, a relatively simple unitary matrices (26) and (28) within the framework of the quantum neural net (17) or(21), allows one to store several different patterns, namely: static patterns and periodically oscillating patterns. This means that in terms of the maximum likelihood dynamics, the quantum neural net behaves as a typical nonlinear system. However, the maximum likelihood dynamics cannot be identified with a deterministic dynamics. Indeed, if one runs Eq. (21) several times, all the solutions may be different from each other, so that with a small probability a pattern may converge to a "wrong" attractor; moreover, a pattern may wander between all five attractors performing a new stochastic paradigm. Strictly speaking such a "leak" from the deterministic performance of the maximum likelihood dynamics is a source of errors in the performance of a neural net. However, in many cases when neural net is expected to display certain flexibility by escaping a prescribed paradigm, this leak may create a useful emergent behavior.

In order to evaluate deviations from the maximum likelihood solution, one has to turn to the probabilistic description of Eqs. (17) and (21).

7. Evolution of Probabilities

Let us take another look at Eq. (17). Actually it performs a mapping of an i^{th} eigenvector into an j^{th} eigenvector:

$$\{0\,0\ 0\underset{\substack{\uparrow\\ i}}{1}0 \quad 0\} \rightarrow \{0\,0\ 0\underset{\substack{\uparrow\\ j}}{1}0 \quad 0\} \tag{29}$$

The probability of the transition (29) is uniquely defined by the unitary matrix U:

$$p_i^j = \left|U_{ji}\right|^2, \quad \sum_{i=1}^{n} p_i^j = 1 \tag{30}$$

and therefore the matrix $\|p_i^j\|$ plays the role of the transition matrix in a generalized random walk which is represented by the chain of mapping (29).

Thus, the probabilistic performance of Eq. (17) has remarkable features: it is quantum (in a sense of the interference of probabilities) in between two consecutive measurements, and it is classical in description of the sequence of mapping (29).

Applying the transition matrix (30) and starting, for example, with eigenstate {1 0...0}, one obtains the following sequence of the probability vectors:

$$\begin{aligned}
&\pi_0 = \{10\ldots0\},\\
&\pi_1 = \{10\ldots0\}\begin{pmatrix} p_1^1 & \cdots & p_1^n \\ \cdots & \cdots & \cdots \\ p_n^1 & \cdots & p_n^n \end{pmatrix} = \{\pi_1^1, \ldots \pi_n^1\}\\
&\pi_m = \{10\ldots\ 0\}\begin{pmatrix} p_1^1 & \cdots & p_1^n \\ \cdots & \cdots & \cdots \\ p_n^1 & \cdots & p_n^n \end{pmatrix}^m
\end{aligned} \tag{31}$$

An i’” component of the vector π_m, i.e., π_m^i, expresses the probability that the system is in the i^{th} eigenstate after m steps.

As follows from Eqs. (31), the evolution of probabilities is a linear stochastic process, although each particular realization of the solution to Eq. (17) evolves nonlinearly, and one of such realization is the maximum likelihood solution considered in the previous section. In this context, the probability distribution over different particular realizations can be taken as a measure of possible deviations from the best estimate solution.

However, the stochastic process (31) as an ensemble of particular realizations, has its own invariant characteristics which can be expressed independently on these realizations. One of such characteristics is the probability $f_{ij}^{(m)}$ that the transition from the eigenstate *i* to the eigenstate *j* is performed in *m* steps. This characteristic is expressed via the following recursive relationships [6]:

$$f_{ij}^{(1)} = p_{ij}^{(1)} = p_{ij,}\ f_{ij}^{(2)} = p_{ij}^{(2)} - f_{ij}^{(1)} p_{jj} \qquad \textbf{(32)}$$

$$f_{ij}^{(m)} = p_{ij}^{(m)} - f_{ij}^{(1)} p_{jj}^{(n-1)} \dot{-} f_{ij}^{(3)} p_{jj}^{(n-2)} \ldots - f_{ij}^{(n-1)} p_{jj}.$$

If

$$\sum_{m=1}^{\infty} f_{ij}^{(m)} < 1 \qquad \textbf{(33)}$$

then the process initially in the eigenstate i may never reach the eigenstate j.

If

$$\sum_{m=1} f_{ii}^{(m)} = 1, \qquad (34)$$

then the i^{th} eigenstate is a recurrent state, i.e., it can be visited more than once. In particular, if

$$p_{ii} = 1 \qquad (35)$$

this recurrent state is an absorbing one: the process will never leave it once it enters.

From the viewpoint of neural net performance, any absorbing state represents a deterministic static attractor without a possibility of “leaks.” In this context, a recurrent, but not absorbing state can be associated with a periodic or an aperiodic (chaotic) attractor. To be more precise, an eigenstate I has a period z (z > 1) if $p_{ii}^{(m)}$ = O whenever m is not divisible by z, anti- $\imath$ is the largest integer with this property. The eigenstate is aperiodic

$$\text{if} \qquad \imath = 1 \tag{36}$$

Another invariant characteristic which can be exploited for categorization and generalization is reducibility, i.e., partitioning of the states of a Markov chain into several disjoint classes in which motion is trapped. Indeed, each hierarchy of such classes can be used as a set of falters which are passed by a pattern before it arrives at the smallest irreducible class whose all states are recurrent.

For the purpose of evaluation of deviations (or "leaks") from the maximum likelihood solution, long-run properties of the evolution of probabilities (31) are important. Some of these properties are known from theory of Markov chains, namely: for any irreducible erogodic Markov chain the limit $p_{ij}^{(m)}$ exists and it is independent of I, i.e.,

$$\lim_{m\to\infty} p_{ij}^{(m)} = \pi_j \tag{37}$$

while

$$\pi_j > 0, \pi_j = \sum_{i=0}^{k} \pi_i p_{ij}, \ j = 0,1,\ldots \ k; \ \sum_{j=0}^{k} \pi_j, \qquad \pi_j = \frac{1}{\mu_{jj}} \tag{38}$$

Here μ_{jj} is the expected recurrence time

$$\mu_{ii} = 1 + \sum_{\ell \neq i} p_{i\ell} \ \mu_{\ell i} < \infty \tag{39}$$

The definition of ergodicity of a Markov chain is based upon the conditions for aperiodicity (36) and positive recurrence (39), while the condition for irreducibility requires existence of a value of m not dependent upon i and j for which $p_{ij}^{(m)} > 0$ for i and j.

The convergence of the evolution (31) to a stationary stochastic process suggests additional tools for information processing. Indeed, such a process for n-dimensional eigenstates can be uniquely defined by n statistical invariants (for instance, by first n moments) which are calculated by summations over time rather than over the ensemble, and for that a single run of the neural net (17) is sufficient. Hence, triggered by a simple eigenstate, a prescribed by n-invariants stochastic process can be retrieved and displayed for the purposes of Monte-Carlo computations, for modelling and prediction of behavior of stochastic systems, etc.

8. Interference of Patterns

In the previous section we have analyzed the simplest quantum neural net (17) whose probabilistic performance was represented by a single-variable stochastic process equivalent to generalized random walk. In this section we will turn to Eq. (21) which describes multi-variable stochastic process and start with the two -measurement architecture. Instead of (29) now we have the following mapping:

$$\frac{1}{\sqrt{2}}\{00\ldots\underset{\substack{\uparrow\\ i_1}}{1}0\ldots\underset{\substack{\uparrow\\ i_2}}{1}0\ldots0\}\to\frac{1}{\sqrt{2}}\{00\ldots\underset{\substack{\uparrow\\ j_1}}{1}0\ldots\underset{\substack{\uparrow\\ j_2}}{1}0\ldots0\} \qquad (40)$$

i.e.,

$$I_1+I_2\to J_1+J_2 \qquad (41)$$

where 1_2,12, J, and J_2 are the eigenstates with the unit 1 is at the $i_1^{th}, i_2^{th}, j_1^{th}$ and j_1^{th} places, respectively. Then the transitional probability of the mappings:

$$p_{i_1 i_2}^{j_1}(I_1+1,\to J_1)=\frac{1}{2}\left|\overset{*}{J}_1 U(I_1+I_2)\right|^2=\frac{1}{2}\left|U_{j_1 i_1}+U_{j_1 i_2}\right|^2 \qquad (42)$$

$$p_{i_1,i_2}^{j_2}(I_1+I_2\to J_2)=\frac{1}{2}\left|\overset{*}{J}_2 U(I_1+I=\frac{1}{2}\right|U_{j_2 i_1}+U_{j_2 i_2}\Big|^2 \qquad (43)$$

Since these mapping result from two independent measurements, the joint transitional probability y for the mapping (40) is:

$$=p_{i_1 i_2}^{j_1 j_2}(I_1+I_2\to J_1+J_2)=p_{i_1 i_2}^{j_1}\,p_{i_1 i_2}^{j_2}= \qquad (44)$$

$$= \frac{1}{4}\left|U_{j_1 i_1} + U_{j_1 i_2}\right|^2 \left|U_{j_2 i_1} + U_{j_2 i_2}\right|^2$$

One can verify that

$$\sum_{j=1}^{n} p^{j}_{i_1 i_2} = 1, \qquad \sum_{j_1 j_2 = j}^{n} p^{j_1 j_2}_{i_1 i_2} = 1 \tag{45}$$

It should be emphasized that the input patterns I_1 and I_2 interfere, i.e., their probabilities are added according to the quantum laws since they are subjected to a unitary transformation in the quantum device. On the contrary, the output patterns J_1 and J_2 do not interfere because they are obtained as a result of two independent measurements.

As mentioned above, Eq. (44) expresses the joint transition probabilities for two stochastic processes

$$I_1 \rightarrow J_1 \text{ and } 1_2 + J_2 \tag{46}$$

which are coupled via the quantum interference (42) and (43):

$$I_1 + I_2 \rightarrow J_1 + J_2 \tag{47}$$

At the same time, each of the stochastic processes (46) considered separately has the transition probabilities following from Eq. (30):

$$I_1 + I_2 \rightarrow J_1 + J_2 \tag{48}$$

and by comparing Eqs. (44) and Eq. (48), one cane see the effect of quantum interference for input patterns.

It is interesting to notice that although the probabilities in Eqs. (44) and (48) have a tensor structure, strictly speaking, they are not tensors. Indeed, if one maps the Hamiltonian H, and therefore, the unitary matrix U to a different coordinate system, the transformations of the probabilities (44) and (48) will be different from those required for tensors. Nevertheless, one can still formally apply the chain rule for evolution of transitional probabilities, for instance:

$$p_{i_1i_2}^{q_1q_2}\left(I_1+I_2\to J_1+J_2\to Q_1+Q_2\right)=p_{i_1i_2}^{j_1j_2}\ p_{j_1j_2}^{q_1q_2}\ \text{etc.} \tag{49}$$

Eqs. (44) and (49) are easily generalized to the case of ℓ measurements $(\ell \leq \text{n})$:

$$p_{i_1\ldots i_\ell}^{j_1\ldots j_\ell}=\frac{1}{\ell^\ell}\prod_{\alpha=1}^{\ell}\left|\sum_{\beta=1}^{\ell}U_{j_\alpha i_\beta}\right|^2, \qquad p_{i_1\ldots i_\ell}^{q_1\ldots q_\ell}=p_{i_1\ldots i_\ell}^{j_1\ldots j_\ell}p_{j_1\ldots j_\ell}^{q_1\ldots q_\ell} \quad \text{etc.} \tag{50}$$

There are two ways in which many-measurement architecture can be implemented: consecutive measurements applied to the same unitary matrix, or by parallel measurements applied to several identical unitary matrices. However, in the last case (see Fig 1) one can introduce different matrices $U^{(1)},\ldots U^{(\ell)}$, and then Eq. (50) can be generalized to:

$$p_{i_1\ldots i_\ell}^{j_1\ldots j_\ell}=\frac{1}{\ell^\ell}\prod_{\alpha=1}^{\ell}\left|\sum_{\beta=1}^{\ell}U_{j_\alpha i_\beta}^{(\alpha)}\right|^2 \tag{51}$$

Now the stochastic processes, in general, are correlated, and the existence of their joint probability cannot be guaranteed [7].

Another useful change in the quantum net architecture based upon pattern interference is the following: assume that the result of the measurement, i.e., a unit vector

$$a_m(t)=\left\{00\ldots0\underset{\overset{\uparrow}{i}}{1}0\ldots0\right\} \tag{52}$$

is combined with an arbitrary complex vector m:

$$m=\{m_1,\ldots m_n\} \tag{53}$$

such that

$$a(t)=[a_m(t)+m]c,\ c=\frac{1}{m_1^2+\dots(m_i+1)^2+\dots m_n^2} \qquad (54)$$

Then the transition probability matrix changes from (30) to

$$p_i^j=\frac{[U_{j1}m_1+\dots U_{ji}(m_i+1)+\dots U_{jn}m_n]}{m_1^2+\dots(m_i+1)^2+\dots m_n^2} \qquad (55)$$

Thus, now the structure of the transition probability matrix p_i^j can decontrolled by the interference vector m.

Eq. (55) is derived for a one-dimensional stochastic process, but its generalization to ℓ -dimensional case is straight-forward

In order to clarify the more complex architectures of quantum neural nets, for instance, such as those given by Eqs. (21), turn to Eqs. (44), and consider the tensor $p_{i_1i_2}^{j_1j_2}$. By simple manipulation of indeces one obtains:

$$p_{i_1i_2}^{j_1j_2}\pi_{j_1}\pi_{j_2}=\pi_{i_1}\pi_{i_2} \qquad (56)$$

The products $\pi_{j_1}\pi_{j_2}$ *and* $\pi_{i_1}\pi_{i_2}$ represent the components of the direct product of two vectors $\pi_1\otimes\pi_2$ and therefore, Eq. (56) can be rewritten as:

$$[\pi_1\otimes\pi_2]_{t+\tau}=p_{12}[\pi_1\otimes\pi_2]_t \qquad (57)$$

where p_{12} is the tensor with the components $p_{i_1i_2}^{j_1j_2}$, and $\pi_1\pi_2$ represents probability vectors for two different stochastic processes coupled via quantum interference (see Eq. (44)).

In order to understand the physical meaning of Eq. (57), start with a simpler case when two stochastic processes $\pi_1\pi_2$ are considered separately. Then each vector evolves according to the following equation:

$$\pi_1(t+\tau)=p_i^j\pi_j(t) \qquad (58)$$

Moreover, if these processes are coupled in a “quantum” sense, one arrives at a simultaneous system:

$$\pi_i^{(1)}(t+7)\text{=/);'7rj, }(t) \qquad \textit{(59)}\ .$$

$$\pi_i^{(2)}(t+\tau)=p_{i_2}^{j_2}\pi_{j_2}(t) \qquad \textbf{(60)}$$

Obviously, the vectors π_1 and π_2 now represent the conditional probabilities.

Thus, due to the quantum interference, the stochastic vectors π_1 (given π_2) and π_2 (given π_1) are correlated. Their direct product $\pi_1 \otimes \pi_2$ which can be associated with the joint probability, evolve linearly according to Eq. (57). (We should notice again that for more complex architectures of the type(51), the joint probability may not exist).

Eq. (57) can be generalized for ℓ -measurement architectures:

$$[\pi_1 \otimes \pi_2 \otimes \ldots \otimes \pi_\ell]_{t+\tau} = p_{12\ldots\ell}[\pi_1 \otimes \pi_2 \otimes \ldots \otimes \pi_\ell]_t \qquad (61)$$

9. Non-Markovian and Nonlinear Processes

The quantum neural nets (17) or (21), with a slight modification, can generate non-Markovian processes which are “more deterministic” because of higher correlations between values of the vector a_i at different times, i.e., between $a_i(t), a_i(t -$ *r),fl,(t–27),* etc.

Indeed, let us assume that each new measurement is combined with the ℓ previous measurements (instead of ℓ repeated measurements). Then Eq. (50) will express the joint distribution of $a_i(t), a_i(t-$?),... etc.

The evolution of these probabilities is described by the equation following from (61):

$$\pi(t)\otimes\pi(t-\tau)\otimes\ldots\otimes\pi(t-\ell\tau)=p_{12\ldots\ell}\{\pi_1(t-\tau)\otimes\ldots\otimes\pi[t-(\ell+t)\tau] \qquad (62)$$

Thus, instead of ℓ -dimensional Markov process in (61), now we have a one-dimensional non-Markovian process of the ℓ th order.

By combining ℓ_1 new measurements with ℓ_2 previous measurements, one can generate an ℓ_1 -dimensional non-Markovian process of the ℓ_2^{th} order.

So far all the stochastic processes Considered above were linear. Now let us assume that along with the Eq. (17) which is implemented by quantum device, we implement (in a classical way) the associated probability equation (5 S). At this point these two equations are not coupled yet. Now turning to Eqs. (52)-(56), assume that the order of the interference vector m is played by the probability vector π. Then Eqs. (17) and (58) take the form:

$$a_i(t+1)=\sigma_1\left\{\sum_j U_{ij}a_i(t)\right\} \qquad \mathbf{(63)}$$

$$\pi_i(t+1)=\sum p_i^j\,\pi_j(t) \qquad (64)$$

where $$a_i(t)=\left[\left\{0\,0\ \ldots 0\,\underset{\underset{i}{\uparrow}}{1}\,0\ \ldots 0\ldots\right\}+\{\pi_1\pi_2\ldots\pi_n\}\right]C \qquad (65)$$

$$C=\frac{1}{\pi_1^2+\ldots(\pi_i+1)^2+\pi_n^2} \qquad (66)$$

$$p_i^j = \frac{U_{j1}\pi_1 + \ldots U_{j_i}(\pi_1 + 1) + \ldots U_{j_n}\pi_n}{\pi_1^2 + \ldots (\pi_1 + 1)^2 + \ldots \pi_n^2} \tag{67}$$

and they are coupled. Moreover, the probability evolution (64) becomes nonlinear since the matrix p_i^j depends upon the probability vector π .

10. Applications of Quantum Neural Nets

There are two broad areas in which classical neural nets become very effective: The associative memory and optimization. In this section we will analyze what additional advantages can be expected from a quantum implementation of neural nets.

The problem of associative memory is formulated as following: store a set of q n-dimensional patterns $\xi_i^\eta (\eta = 1, 2, \ldots q; i = 1,2,\ldots n)$ as dynamical attractor; if a new pattern ξ_i presented as an input is sufficiently close to a particular pattern $\bar{\xi}_i^\eta$, i.e., it belongs to the basin of the corresponding attractor, it will trigger a dynamical process which eventually converges to the sample pattern $\bar{\xi}_i^\mu$. From the viewpoint of information processing, such a convergence can be interpreted not only as associative memory, but also as pattern recognition, identification, classification etc. However, the most important part of this process which distinguishes neural nets from other computational tools is generalization. Indeed, the convergence of the solution to an attractor is a dissipative process: it is accompanied by the loss of (unnecessary) information. Only invariants which characterize the belonging of a pattern to a certain class survives this loss, and they are represented by the attractor.

The fundamental problem in associative memory is to find such a synaptic interconnections T_{ij} (see Eq. 1), or, in case of a quantum implementation, the Hamiltonian H, which provides a prescribed number of attractors of certain type and at certain locations.

In optimization performance the problem is inverse: the matrix T_{ij} (or H_{ij}) is given, and the neural net must converge to an attractor which represent a minimum to a certain function (or functional) formulated in terms of the matrices T_{ij} or H_{ij} .

There are several advantages which can be expected from quantum implementation of neural nets. Firstly, since the dimension of the unitary matrix n can be exponentially larger within the same space had it been implemented by a quantum device, the capacity of quantum neural nets in terms of the number of patterns stored as well as their dimensions can be exponentially larger too.

Secondly, quantum neural nets have a new class of attractors representing different stochastic processes, which in terms of associated memory, can store complex behaviors of biological and engineering systems, or in terms of optimization, to minimize a functional whose formulation includes statistical invariants.

But the most remarkable property of quantum neural nets is associated with the pattern interference. Indeed, let us assume that we store letters of the alphabet in the form of the corresponding stochastic attractors ξ_{η}. Then if some of these letter, say f,,,... ξ_{η_ℓ}, are presented to the neural net simultaneously, their processing will be accompanied by quantum interference in such a way that they will converge to a new attractor, say $\xi_{1,2..\ell}$. This new attractor preserves the identities of the letters $\xi_{\eta_1 \ldots} \xi_{\eta_\ell ?}$, but at the same time, it is not a simple sum of these letters. Moreover, any additional letter $\xi_{\eta_{\ell+1}}$ will create a totally different new attractor $\xi_{1 \ldots \ell, \ell+1}$. Actually this phenomenon is similar to formation of words from letters, sentences from words, etc. In other words, the pattern interference creates a grammar by giving different meaning to different combinations of letters. However, although this grammar is imposed by natural laws of quantum mechanics, it can be changed. Indeed, by changing phases of the components H_{ij} of the Hamiltonian, one changes the way in which the patterns interfere and therefore, the "English" grammar can be transformed into "French" grammar.

A. Learning

In order to achieve a required performance of a neural net, one has to assign appropriate values to the synaptic weights $7_{[}$, i.e., to solve an inverse problem for Eq. (1). Since the number of the weights can be enormously large, in classical neural nets analytical inversion is replaced by an equivalent dynamical procedure called learning.

In order to illustrate how learning can be applied to quantum neural net, let us turn to the simplest case (17), and rewrite Eq. (31) for probability evolution in the form:

$$\left\{\pi^{m}_{(1)} \ldots \pi^{m}_{(n)}\right\} = \left\{\pi^{0}_{(1)} \ldots \pi^{0}_{n}\right\} \begin{pmatrix} p^{1}_{1} & \cdots & p^{n}_{1} \\ & & \\ p^{1}_{n} & \cdots & p^{n}_{n} \end{pmatrix} \qquad (68)$$

where $\pi^{0}_{(i)}$ and $\pi^{m}_{(i)}$ are the components of the input and output vectors, respectively, and m is the number of computational steps.

Let us require that in response to the probabilistic input $\pi^{0}_{(i)}$, the quantum neural net must converge to a stationary stochastic process with the probability distribution $\pi^{m}_{(i)}$. Moreover, assume that there are n different input vectors $\pi^{m}_{(ij)}$ Then Eq. (51) can be presented in the following matrix form:

$$\begin{pmatrix} \pi^{m}_{11} & \cdots & \pi^{m}_{(1n)} \\ \cdots & \cdots & \cdots \\ \pi^{m}_{(n1)} & \cdots & \pi^{m}_{nn} \end{pmatrix} = \begin{pmatrix} p^{1}_{1} & \cdots & p^{n}_{1} \\ \cdots & \cdots & \cdots \\ p^{1}_{n} & \cdots & p^{n}_{n} \end{pmatrix}^{m} \begin{vmatrix} \pi^{0}_{(11)} & \cdots & \pi^{0}_{(1n)} \\ \cdots & \cdots & \cdots \\ \pi^{0}_{(n1)} & \cdots & \pi^{0}_{(nn)} \end{vmatrix} \qquad (69)$$

whence by inversion:

$$\begin{pmatrix} p^{1}_{1} & \cdots & p^{n}_{1} \\ \cdots & \cdots & \cdots \\ p^{1}_{n} & \cdots & p^{n}_{n} \end{pmatrix} = \lim_{m \to \infty} \left[\begin{pmatrix} \pi^{m}_{(11)} & \cdots & \pi^{m}_{(1n)} \\ \cdots & \cdots & \cdots \\ \pi^{m}_{(n1)} & \cdots & \pi^{m}_{(nn)} \end{pmatrix}^{m} \begin{vmatrix} \pi^{0}_{(11)} & \cdots & \pi^{0}_{(1n)} \\ \cdots & \cdots & \cdots \\ \pi^{0}_{(n1)} & \cdots & \pi^{0}_{(nn)} \end{vmatrix} \right]^{m} \qquad (70)$$

Eq. (70) defines the sought transition matrix only when the limit exists, i.e., when the corresponding Markov chain is irreducible and ergodic. The last two requirements impose some constraints upon the input-output relationships in Eq. (69). For the purpose of illustration of the learning strategy, we will assume that the input-output relationships are assigned in such a way that these constraints are already satisfied, and therefore, Eq. (70) presents a unique solution to the problem of finding the transition matrix for the prescribed performance of the neural net.

Now invoking Eq. (30) one finds all the elements of the corresponding unitary matrix U_{ij} to the accuracy of the phases which can be set up arbitrarily. It should be noticed that the phase invariance of the unitary matrix is not the rule: it is a result of simplicity of the chosen neural net. Indeed, in case (21) the relationships between the transitional probabilities and the elements of the unitary matrix include phases (see eq. (44)).

Thus, the assignments for the elements of the unitary matrix were found in analytical form, and in a relatively simple way. Unfortunately, it is still not very useful. Indeed, even in classical neural nets the number of synaptic weights is so large that their progratnming based upon analytically found values is unthinkable. That is why such a programming is replaced by learning based upon dynamical convergence of synaptic weights to the correct values. First we would like to introduce a surprisingly simple way to generate a sufficiently rich set of unitary matrices out of a change of only one parameter: the time τ between two consecutive resets.

Let us start with Eq. (10), and rewrite it for the unitary matrix at the end of the simple quantum computation period τ applying the Sylvester decomposition:

$$U^{(\tau)} = e^{iH\tau/\hbar} = \sum_{k=1}^{n} e^{i\lambda_k \tau/\hbar} \frac{\prod_{j\neq k}\left(H - \lambda_j I\right)}{\prod_{j\neq k}\left(\lambda_k - \lambda_j\right)} \tag{71}$$

where H is the Hamiltonian of the quantum system, and $\lambda_j (j = 1,2. . . n)$ are its eigenvalues, while

$$I_m \lambda_j = 0, \qquad \lambda_j \neq \lambda_\ell \text{ if } j \neq \ell \tag{72}$$

As follows from Eq. (71), each component of the unitary matrix U_τ for any fixed Hamiltonian is a sum of n periodic functions with periods 2 π/λ_k z. If the eigenvalues λ_j do not commensurate, the behavior of the unitary matrix U_τ as a function of τ, will be ergodic, i.e., this matrix eventually will take all possible values which result from all the possible combinations of its n eigenvalues' $e^{i\lambda_j \tau/\hbar}$. In other words, with no changes in the Hamiltonian H, one can arrange a set of unitary matrices which is equivalent to those obtained from variations of n independent parameters.

In order to find the optimal period τ, start with the following Lyapunov function:

$$L = \frac{1}{2}\sum_{i,j=1}^{n}\left(p_i^j - \left|U_{ji}\right|^2\right)^2 \qquad (73)$$

Comparing Eq. (73) and Eq. (30), one finds that Eq. (30) is satisfied when L has its minimum. The dynamical system which converges to this minimum can be written in the form:

$$\frac{d\tau}{dt} = -\frac{\partial L}{\partial \tau} = \sum_{i,j=1}^{n}\left(p_i^j - \left|U_{ji}\right|^2\right)\frac{\partial\left|U_{ji}\right|^2}{\partial \tau} \qquad (74)$$

In Eqs. (56) and (57), the parameters p,’ are obtained from Eq. (53). The explicit expressions for $U_{,,l}(\tau)^2$ and ∂ U $_{ji}^{\,2}\big/\partial\tau$ can be found from Eq. (71).

It should be noticed that the Lyaunov function (73) is not quadratic with respect to the sought parameter τ, and therefore, it can have more than one minimum. However, for a one-dimensional case, finding a global minimum is not a hard problem.

In order to utilize the full capacity of quantum neural nets, in addition to optimal τ, one has to find and implement optimal orientation of the Hamiltonian, and that may be costly. However, quantum interference offers an alternative way for learning which complements the approach described above without changing the Hamiltonian itself. Indeed, let us turn to the paradigm described by Eqs. (52)-(55). As follows from Eq. (55), one can choose an interference vector m to incorporate 2 $n - 1$ additional free parameters into the transition probability matrix $p,^{;}$, which thereby, will have total $3n - 1$ parameters. Now one has to choose these parameters such that the stochastic process will converge to a prescribed limit probability distribution π^*

The actual way to do this is the following. If π^* is the limit probability distribution, then the following equations must be satisfied:

$$\sum_{j-1}^{n} p_i^j \pi_j^* = \pi_i^*, \quad \sum_{i=1}^{n} \pi_i^* = 1, \quad \sum_{j=1}^{n} p_i^j = 1 \qquad (75)$$

Since the prescribed probability vector π^* must be normalized the matrix p_i^j defined by Eq. (55) is normalized in advance, the first constraint in Eq. (75) is satisfied. But since the vector (52) is also normalized as well, and therefore, the second constraint in (75) is satisfied too. But then as follows from Eq. (75)

$$\sum_{j=1}^{n} \sum_{i=1}^{n} p_i^j \pi_j^* \equiv \sum_{i=1}^{n} \pi_i^* = 1, \qquad (76)$$

i.e., the number of independent equations in the system (75) is n-1.

Thus, only n-1 constraints are imposed upon 3n- 1 free parameters of the transition probability matrix p$_{s}(m_{k,}U_{\alpha\beta})$.

This redundancy can be exploited in several ways. Firstly, one can minimize the time of convergence to the stochastic attractor by maximizing the determinant

$$\max\left|\det\left(p_i^j - \delta_i^j\right)\right| \qquad (77)$$

subject to the unilateral constraint:

$$p_{ij} \geq 0 \qquad (78)$$

Here δ_i^j is the Kronecker delta.

Secondly, one can exploit the entanglement effect by presenting the original n x n unitary matrix as a tensor product of q **2** x **2** matrices:

$$U = U_1 \otimes U_2 \otimes \ldots \otimes U_{q.} \tag{79}$$

The interference vector can be presented in the same way:

$$m = m_1, \otimes m_2 \otimes \ldots \otimes m_q \tag{80}$$

while

$$n = 2^q \tag{81}$$

Then the total number of free parameters in the matrix p_i^l *is* 5q i.e., **5** $\log_2$ **n.** Therefore, the limit stochastic process $\pi*$ can be approximated to accuracy of 5 $\log_2 n$ parameters out of total n – 1 parameters required for its full description. For many practical cases such an approximation is sufficient. Due to the entanglement effects (79) and (80), this approximation is achieved by means of exponentially smaller resources.

Of course the price we pay for using unitary matrices and initial state vectors that have a direct product structure is that we exclude the majority of possibilities (i.e. arbitrary unitary matrices and arbitrary, entangled states). However, there is a sound pragmatic reason for focussing on objects which have a direct product structure: they will probably be much simpler to implement physically and yet still give acceptable approximations to the ideal behavior. Nevertheless, if efficient technologies can be developed that can actually create arbitrary unitary operators and arbitrary states then our model will be applicable to these systems too.

The actual learning procedure, i.e., finding the interference m-vector from prescribed probability distribution π^* (at a fixed unitary matrix U) can be based upon the gradient descend procedure well established in theory of neural nets:

Start with the "energy" function to be minimized:

$$E = \frac{1}{2}\left[\left(p_{21}\pi_1^* + \ldots p_{2n}\pi_n^* - \pi_2^*\right)^2 + \ldots + \left(p_{n1}\pi_1^* + \ldots + p_{nn}\pi_n^*\right)^2\right] \rightarrow \min \qquad (82)$$

Then the interference vector m_i is found as the static attractor of the following system:

$$\frac{dm_i}{dt} = -\frac{\partial E}{\partial p_{ij}}\frac{\partial p_{ij}}{\partial m_i} \qquad (83)$$

while the derivatives $\partial p_{ij} / \partial m_i$ are found from Eqs. (55). Since (82) plays the role of a Lyapunov function, a solution to Eqs. (83) always exists (but it can be non-unique). Thus, quantum learning is based upon two fundamental quantum effects: Entanglement and interference. Due to entanglement, any prescribed stochastic attractor can be approximated with a sufficient accuracy by means of exponentially smaller resources. Due to interference, each initial vector "mixed" with the corresponding interference vector, triggers a stochastic process which converges to a prescribed probability distribution. (It should be recalled that without such an interference, all initial vectors would converge to the same stochastic attractor because the governing equations for the probabilities are linear). .

B. Associative Memory

Quantum learning opens up a very simple way for implementation of associative memory, as well as of pattern recognition and classification. Indeed, consider Eqs. *(75)*

which relate the interference vector m and the limit probability distribution π^*, and require that

$$\pi_i^* = 0 \; if \quad i < i_{1,} \;\; or \; i > i_2, \;\; i_2 > i_1 \qquad (84)$$

this means that the stochastic process with the probability 1 will approach a selected domain in between i_1 and i_2. The number of constraints imposed by Eq. (84) upon the interference vector m is:

$$\gamma = i_1 + i_2 < n \qquad \textbf{(85)}$$

where n is the dimensionality of π^*.

Therefore, there exists a $(2n - 1 - y)$ -parametrical family of vectors m which belongs to the domain (84) in a sense that each stochastic process triggered by a unit vector "mixed" with the interference vector m from this family will conerge to the domain (84).

Thus, solutions to Eqs. (75) subject to the constraints (84) establishes a correspondence between the" class of patterns (represented by the family of m-vectors) and the domains in the n-dimensional space to which the stochastic processes (triggered by m-vectors) converge. This procedure can be exploited for associative memory, pattern recognition, and categorization.

C. Optimization

As mentioned earlier, neural networks are expected to solve some optimization problems by finding a configuration which minimizes some functional usually referred as an energy function.

In order to find an energy function which is minimized by the simplest neural net (17), consider the corresponding evolution of probability(31):

$$p_i(t + \tau) = \sum p_i^j \pi_j \,(t) \qquad \textbf{(86)}$$

where p_i^j *is* expressed by Eqs. (30), and τ is the period of one iteration.

The solution to Eq. (86) is stable if

$$|\lambda_i| \leq 1 \qquad \textbf{(87)}$$

where λ_i are the characteristic roots of the transitional probability matrix p_i^j.

In order to proof (87), consider the second norm of the matrix p_i^j:

$$\text{p,}^{\text{J}} = \text{max} \sum_{i=1}^{n} \left| p_i^j \right| = 1$$

(88)

Hence:

$$\max_i \lambda_i | \leq \| p_i^j \|_2 = 1 \qquad (89)$$

Assuming that τ is sufficiently small in a sense that

$$\pi_1(t + \tau) - \pi_i(t) = \frac{d\pi_i}{dt} \tau \qquad \mathit{(90)}$$

rewrite Eqs. (86) as

$$\frac{dp_{i}}{dt} = b_{ij}\; p_{j} \text{ where } b_{ij} = \left(p_i^j - \delta_{ij}\ \right) \tag{91}$$

or in a more general form:

$$\frac{dp_i}{dt} = b_{ij} p_j + \gamma\left(1 - \sum_{i=1}^{n} p_i\right) \tag{92}$$

where γ is an arbitrary multiplier.

Eqs. (92) is equivalnt to Eq. (91) because the expression in the brackets is zero (See Eqs. 30).

If the matrix b_{ij} is symmetric:

$$b_{ij} = b], \tag{93}$$

one can find such a quadratic form (with respect to the variables π_1) that Eq. (92) is a grdient system for it. Indeed, consider a quadratic form

$$E_o = \frac{1}{2} \sum_{i,j=1}^{n} b_{ij}\ \pi_i\ \pi_j \tag{94}$$

which is supposed to be minimized subject to the following constraint:

$$\sum_{i=1}^{n} p_{,} = l \tag{95}$$

In other words, one has to find an unconstrained minimum for the function:

$$E = \frac{1}{2}\sum_{i,j=1}^{n} b_{ij} p_i p_j + \frac{\gamma}{2}\left(1 - \sum_{i=1}^{n} p_i\right)^2 \to \min \tag{96}$$

The conditions for the minimum are:

$$\frac{\partial E}{\partial p_i} = B_{ij} p_j + \gamma\left(1 - \sum_{i=1}^{n} p?_i\right) = 0 \tag{97}$$

$$b_{11} > 0 \qquad \begin{vmatrix} b_{11} & b_{,2} \\ b,_{2} & b_{22} \end{vmatrix} > 0, \cdots \det b_{ij} > 0 \tag{98}$$

But the solution (97) is a dynamical attractor for the system of differential equation (92): indeed, the system is stable (because of the inequalities (98)), and it has only one attractor (97). One can also verify that Eq. (96) plays the role of a Lyapunov function for the system (92).

Thus, the quantum neural net (17) minimizes the quadratic functional (94) subject to the constraints (95). This minimization should be understood in the following way: The solution to Eqs. (17) converges to a stationary (ergodic) stochastic process which uniquely defines the probability vector

$$\pi_1 = \lim_{t \to \infty} \pi_i(t) \tag{99}$$

i.e., the solution to the problem.

One should note that the computational problem (94), (95) is posed in such a way that it has some specific features, namely the coefficients b_{ij} of the quadratic form must satisfy the conditions

$$\sum_{j=1}^{n} b_{ij} = 0 \tag{100}$$

which follow from Eqs. (30) and (91).

Actually Eqs. (100) enforce the constraints (95) automatically if the initial vector $\pi_i(0)$ satisfies them, i.e., if

$$\sum_{i=1}^{n} \pi_i(0) = 1 \tag{101}$$

Obviously, the constraints (100) restrict the class of computational problems which can be solved by the quantum neural net (17). However, if we move to more complex architectures (see Eqs. (21)-(23)), the restrictions (100) can be eliminated. Moreover, even slight changes in the simplest architecture (17) can do it. Indeed, so far the reset values of the probability vector were identical to its measurement values. Let us now assume that no matter what the measurements are, the last components of the reset vector is always zero. In other words, we have created a leak of probability. Then instead of the constraint (95) one has

$$\sum_{i=1}^{n} \pi_i < 1 \qquad (102)$$

and the restriction (100) can be dropped.

It should be noticed that the inequalities (98) do not restrict a computational problem since they must be satisfied for any problem which is expected to have a minimum. On the first sight, the problem which has been discussed above is one of the simplest one. However, the main difficulty occurs when the dimensionality of the problem becomes exponentially large. Then one can face a typical NP-complete situation when it is very easy to check the solution, but the number of such checks to find it is exponentially large. It is not a coincidence that such hard problems as the famous traveling salesman problem, the weighted matching problem, or the graph bipartitioning problem can be reduced to finding the minimum of a quadratic form subject to linear constraints. In this respect, the main advantage of quantum neural nets is in their ability to arrange a dynamical attractor of exponential dirnensionality in a “polynomial” space. The number of iterations necessary for approaching this attractor (they can be taken as an equivalent of “digital” complexity) will not grow exponentially with the growth of the dimensionality because of the dynamical parallelism of evolution along each dimension.

For a two-measurement architecture, the energy function minimized by Eqs. (57) is:

$$H = \frac{1}{2} p_{i_1 i_2}^{j_1 j_2} \ \pi_{i_1} \pi_{i_2} \pi_{j_1} \pi_{j_2} = \frac{1}{2} p_{j_1 j_2}^{i_1 i_2} \ \pi_{j_1} \pi_{j_2} \pi_{i_1} \pi_{i_2} = \cdots \qquad \text{etc} \qquad (103)$$

As follows from (103) the tensor p_{12} must be symmetric with respect to permutations of all its indices:

$$p^{j_1 j_2}_{i_1 i_2} = p^{j_1 j_2}_{j_1 j_2} = p^{i_1 i_2}_{j_1 j_2} = p^{j_2 i_1}_{i_2 j_1} = \cdots \qquad \text{etc} \qquad (104)$$

In addition to that, its components must satisfy the constraints (95)

$$\sum_{j_1 1_2}^{n} p^{j_1 j_2}_{i_1 i_2} = 1 \quad \text{for} \quad i_1\, i_2 = 1, 2 \cdots n \qquad (105)$$

It should be recalled that the tensor p_{12} is uniquely defined by the unitary matrix U via Eqs. (44).

For an ℓ -measurements architecture, the energy function minimized by Eqs. (6 1) is:

$$H = \frac{1}{2} p^{j_1 \cdots j_\ell}_{i_1 \cdots i_\ell} \quad \pi_{i_1} \ \pi_{j_1} \cdots \pi_{j_\ell} \qquad (106)$$

while the tensor $p_{12 \ldots \ell}$ must be symmetric with respect to permutations of all its indices, i.e.,

$$p^{j_1 \ldots j_\ell}_{i_1 \ldots i_\ell} = p^{i_1 \ldots i_\ell}_{j_1 \ldots j_\ell} = \cdots \qquad \text{etc} \qquad (107)$$

and its component must satisfy the constraint

$$\sum_{j_1,\dots j_\ell=1}^{n} p_{i_1\dots i_\ell}^{j_1\dots j_\ell} = 1 \; for \; i_{1,}\dots i_\ell \quad \mathbf{1,2\dots 71} \tag{108}$$

As in the two-dimensional architecture, the tensor $p_{12..}$ *is* uniquely defined by the unitary matrix U via Eqs. (50).

Let us make some comments on the capacity of the ℓ -measurements quantum neural nets. For that purpose we will assume that instead of ℓ sequential measurements performed on the same quantum device, we have ℓ parallel identical quantum devices which allow one to perform all the measurements simultaneously. Then, loosely speaking, the space occupied by such a device will be:

$$\mathrm{S}, \propto n\ell \tag{109}$$

Although the number of degrees of freedom, i.e., the number of equations in the system (61) is still equal to $n\ell$, the number of the components in the quadratic form (106) is

$$S_2 = \sum_{k=1}^{\ell} C_n^k = \sum_{k=1}^{\ell} \frac{n!}{k!(n-k!)} = 2^n \qquad \text{if } \ell = n \tag{110}$$

where C_n^k are the binomial coefficients. Obviously, with the growth of the dimensionality n, S_2 grows exponentially faster than S_1, and this leads to exponential “compression” of space occupied by the ℓ -- measurements architecture of quantum neural nets. One has to recall that this “compression” should be added to the original “compression” performed by quantum entanglement in the unitary devise according to which

$$n \sim 2^q, \text{ i.e., } S_2 \ \text{-}2^{2}{}' \tag{111}$$

where q is the number of “classical” degrees of freedom.

But since the capacity of neural nets, loosely speaking, is linearly proportional to the number of its degrees of freedom, one can conclude that ℓ -measurement quantum neural nets possess an enormous capacity in comparison to classical neural nets.

11. Universal generator of stochastic processes

As shown above, the quantum neural nets can be viewed as a universal and compact generator of stochastic processes, that cannot be achieved, even in principle, by *any* classical device. Indeed, it can generate multi-variate correlated or non-correlated, Markovian and non-Markovian, linear and nonlinear stochastic processes with prescribed properties by simply changing a quantum interference pattern without even touching the quantum hardware, i.e., the Hamiltonian. Due to quantum entanglement, the quantum neural net can be implemented by utilizing exponentially smaller resources.

One of the most important application of simulated stochastic processes is the Monte-Carlo methods discussed in the Introduction. The second area of application is performance of sampling experiments on the model of the systems. In this area not only the limit probability distributions, but their time evolutions are important as well. And in this connection, the nonlinear stochastic processes which allows one to control the current strategy in real time by changing the stochastic attractors and concentrating probabilities in a certain domain’ (depending upon a changing objective) become very useful in modeling decision making process in a game-type situation.

12. Numerical simulations

In this section we will illustrate some of the basic concepts of quantum neural nets by numerical simulations. They illustrate the behavior of a quantum neural net being used to simulate an arbitrary Markov process. The quantum state fed into the QNN at each iteration evolves according to the rule:

$$\left|\psi^{next}\right\rangle = \left\| M\left\{U\left|\psi^{last}\right\rangle + \left|\psi^{init}\right\rangle\right\}\right\|$$

where ψ^{init}) is the initial state vector supplied to the QRN, $\psi^{last}\rangle$ is the last state vector supplied to the QNN, ψ^{next}) is the next state vector that will be supplied to the QNN, U is the unitary matrix that defines the virtual connectivity matrix of the QNN, $M\{\,.\,..\}$ is a measurement operator that projects the state of $U\psi^{last}$) into some eigenstate of M and $\|\|$ denotes renormalization.

The sequence of measurement outcomes $M\left\{U\left|\psi^{last}\right\rangle\right\}$ defines a Markov process with transition probability matrix can be computed exactly. For example, consider the 4x 4 unitary matrix U defined by:

$$\begin{pmatrix} -.426364-.40965\,i & .152799+.449573\,i & .268873-.52106\,i & .262525-.110547\,i \\ .187355+.256512\,i & .377974-.0919836\,i & .624798-.282139i & -.5189+.0924264\,i \\ .478001-.334466\,i & .230266-.310334\,i & .0028144-.0982155\,i & .164187-.688263\,i \\ -.415635+.190776\,\mathrm{i} & --.263578-.635931\,i & .419877-.0166754\,i & .363206-.0920733\,i \end{pmatrix}$$

and let the initial state ψ^{init}) be:

$$\psi^{init}\rangle = \begin{pmatrix} -.00741589+.48916\,i \\ -.12314-.667601\,i \\ .344441-.149759\,i \\ .386876-.095256\,\mathrm{i} \end{pmatrix},$$

The transition probability matrix for this QNN (defined by U and ψ^{init})), which specifies the probability of obtaining the measurement outcome $"j"$ given that the last measurement outcome was “i” is given by:

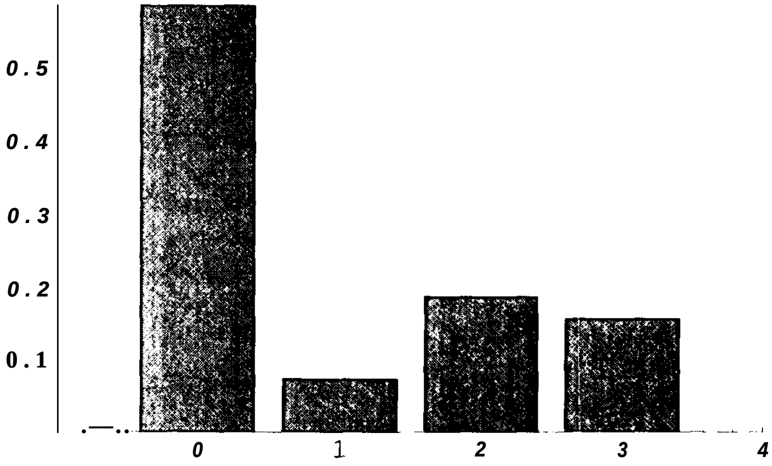
$$\begin{pmatrix} \mathbf{.579626} & \mathbf{.00222459} & \mathbf{.216761} & .201389 \\ .338022 & .0781686 & .156848 & .426962 \\ .785018 & .159218 & .0349651 & .0207996 \\ .471082 & .225005 & .28342 & .0204926 \end{pmatrix}$$

This is certainly a true transition probability matrix as all elements are between O and 1 and the sum of the elements along any row is 1.

In this QNN, there are four possible measurement outcomes which we can label arbitrarily as “O” through “3”. These four outcomes correspond to the four binary values 00, 01, 10, 11 that can be obtained by making a measurement on the state that is generated after the unitary evolution i.e. by measuring the state $U\psi^{last}\rangle$. The QNN defined by U and ψ^{init}) is predicted, according to our theory, to have the fixed point probability distribution over states given by:

$$(.584103 \quad .0719151 \quad .188539 \quad .155444)$$

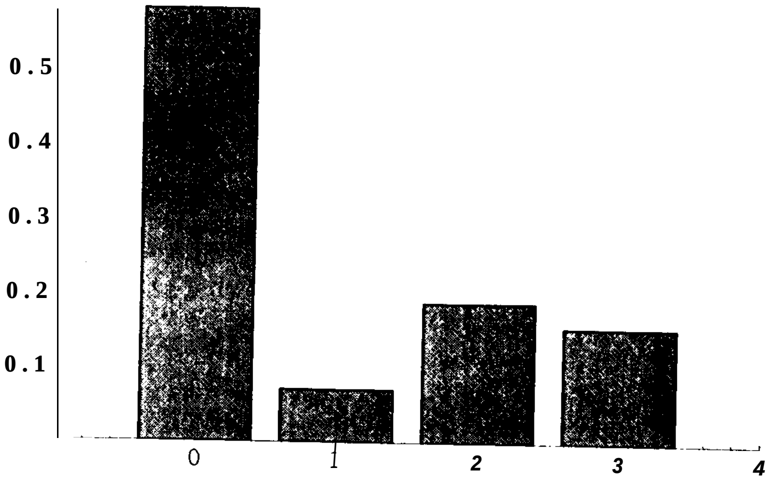
That is we expect to see the measurement outcome “O” 0.584 of the time, the measurement outcome ‘1”, 0.072 of the time etc. Graphically, this distribution is:



We can see that a simulation of a QRN generates a stochastic attractor that does indeed approach this distribution: Here is an actual sequence of 100 measurement outcomes for the **QRN** defined by U and ψ^{init}):

{o,2,o,o,2,o,2,o,o,2,o,o,o,o,o,3,o,o,o,2,o,o,2,o,2,o,o,o,o,o,o,o,o,o,3,2,o,2,o,3,o,3,2,o,o,2,o, 2,0,2,0,2,0,0,0,0,0,2, 1,0,2,0,0,0,0,0,3,2,3,2, *1,3,2,0,0,3,1,3, 1,3,0,3,0,2,0,0,0,2,0,0,3,0,2,0,2, 0,0,0,0,2,2}*

Visually, the distribution of states in this experimental run is found to be :

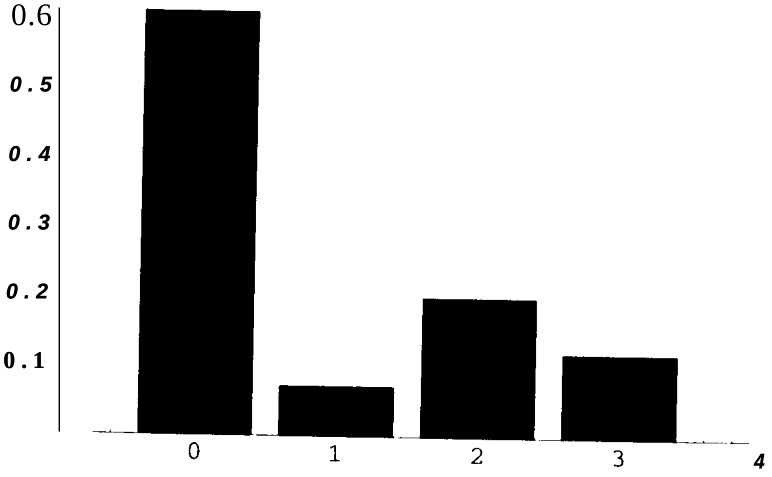


,

Notice the similarity to the predicted distribution. With longer runs, the agreement with the predicted distribution becomes exact. Moreover, a separate run of the same QRN (same U and ψ^{init})) yields a different sequence of measurement outcomes.

{o,o,o,o,o,o,3,o,o,o,2,o,o,o,o,2,o,o,3,2,o,o,o,o,o,2,3,o,o,o,o,o,3,2, 1,2,0,3,1,0,2,0,2,0,2,0,0, 0,0,3,2,0,2,0,0,0,2,0,0,3, 1 ,3,0,0,0,0,0,0,0,3,1,1 ,2,0,0,0,0,2,0,2,0,2,0,0,0,0,2,0,2,0,3,0,0,0,0, 3,2,1 ,2,3,1 }

Nevertheless the sequence converges to the same stochastic attractor as before:



13. Conclusion

Thus, it has been demonstrated that quantum recurrent net as an analog device can be based upon a sequence of quantum and classical computations. During the quantum regime, a stochastic input pattern is transformed (according to Schrodinger equation) into the output stochastic pattern of the same dimensionality. During the following classical regime which includes quantum measurements and reset, the stochastic pattern is contracted into a pattern of lower dimensionality, and this contraction is equivalent to the performance of a sigmoid function. The combined effect of the alternating quantum and classical computations can be described by generalized random walk, i.e., by Markov chains in the form of the Chapman-Kolmogorov equation. Eventually the output pattern approaches an attractor (which can be static, periodic, or ergodic), and such attractors can be utilized for storing certain patterns. The assignment of an appropriate unitary matrix can be based upon the optimal choice of the time period of the regime of quantum computations which actually represents the procedure of learning. But in addition to that the transition probability matrix can be controlled by combining the output vector with an appropriately chosen interference vector.

Let us now summarize advantages and limitations of quantum neural nets.

The most obvious advantage of quantum neural nets which actually gave the motivation for the whole effort is an exponential increase of their capacity due to quantum entanglement. However, the price paid for this is a significant slow down of the convergence to attractors because of measurements and resets which must be performed after each quantum computational step.

A less obvious, but much more fundamental advantage of quantum neural nets is an interference between stochastic inputs as a result of quantum superposition. Due to this interference, the stored patterns acquire a logical structure in a sense that each combination of patterns has a qualitatively new meaning in the same way in which combinations of letters forming words do. This property has a very interesting philosophical consequence. Indeed, it was always difficult to understand how biological neural nets can learn patterns of external world without any preliminary structure built-in to their synaptic interconnections. The experience with artificial neural nets shows that training without a preliminary structure is exponentially longer than those with a structure, and that poses the following question: who created the "first" structure in biological neural nets which provides the ability to learn and select useful properties in polynomial time? In other words, can natural selection act without a "creator"? The quantum neural nets may give a positive answer to this question: the logical structure of synaptic interconnections can be imposed by natural laws of physics, and in particular, by quantum mechanics. Hence, if biological neural nets utilize quantum effects in their performance, they can learn the model of the external world, including its logical structure, in polynomial time without any preliminary structure.

The problems of hardware implementations of quantum devices have not been discussed in this paper. However since the quantum nets operate by interleaving quantum evolution with measurement and reset operations, they are far less sensitive to decoherence than other designs of quantum computers.

"The research described in this paper was performed by the Center for Space Microelectronics Technology, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the National Aeronautics and Space Administration, Office of Space Access and Technology."

1. R. Feynman, Int. J. of Theoretical Physics, Vol. 21, No. 6/7, 1982.

2. J. Taub and H. Wozniakowski, "Breaking Intractability," Scientific American, August 1992, Vol. 46.

3. D. Deutsch, Proc. R. Sot. London A425, 73 (1989).

4. Seth Lloyd, "A Potentially Realizable Quantum Computer," Science, 261: 1569-1571, Sept. 17, 1993.

5. J. Hertz, A. Krogh, and R. Palmer, "Introduction to the Theory of Neural Computation," Addison-Wesley Publishing Co., 1991.

6. M. Bartlett, "An Introduction to Stochastic Processes," Cambridge University Press, 1956.

7. M. Zak, "Dynamical Simulations of Probabilities," Chaos, Solitons, and Fractals, Vol. 8, No. 5, pp. 793-804, 1997.

8. C. P. Williams & S. H. Clearwater, "Explorations in Quantum Computing," TELOS/Springer-Verlag, ISBN: 038794768X(1997)

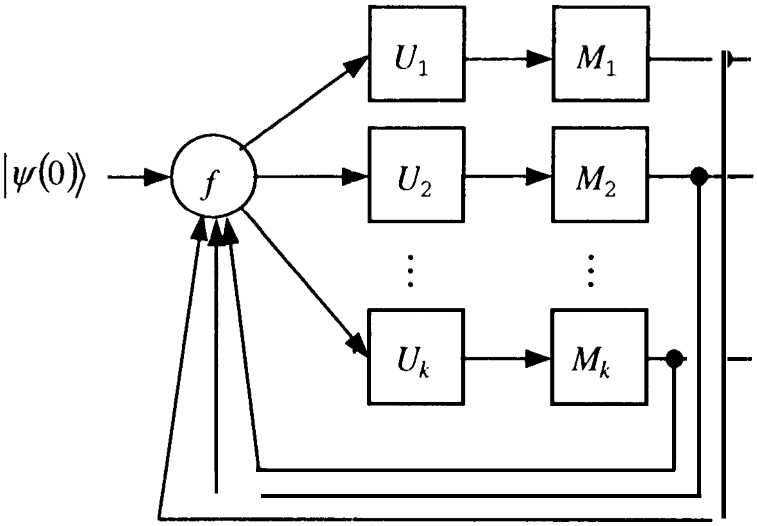


Fig. 1 Thc k-Parallel Quantum Neural Network Architecture

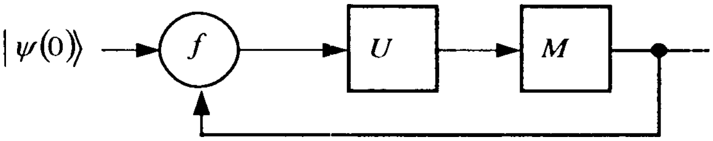


Fig. 2 Thc QNN for Generating a Onc Dimensional Stochastic Attractor